REMARKS/ARGUMENTS

In the Office Action, the Examiner noted that claims 1-19 and 21-24 are pending in the application. The Examiner additionally stated that claims 1-19 and 21-24 are rejected. By this amendment, claims 1, 6, 11, 14, 18, and 21 have been amended. Hence, claims 1-19 and 21-24 are pending in the application.

Applicant hereby requests further examination and reconsideration of the application, in view of the foregoing amendments.

In the Specification

The Examiner noted that Applicant's amendment failed to comply with revised 37 CFR 1.121, and pointed out that in the future, a notice of non-compliant amendment would be sent out for failure to comply. Applicant appreciates the Examiner's forbearance and presumes that all changes to the claims and specification were entered. Accordingly, this amendment 1) complies with revised 37 CFR 1.121 and 2) is based upon changes entered as a result of the previous amendment.

Applicant has thus amended the specification to secure a substantial correspondence between the claims amended herein and the remainder of the specification. No new matter is presented.

In the Claims

Rejections Under 35 U.S.C. §102(b)

The Examiner rejected claims 1-19 and 21-24 under 35 U.S.C. 102(b) as being taught by Bosshart, U.S. Patent Number 5,235,686 (hereinafter, Bosshart). Applicant respectfully traverses the Examiner's rejections.

Prior to providing a claim-by-claim analysis, a brief summary of the teachings of Bosshart are provided to aid the Examiner in her reconsideration of the claims.

Bosshart teaches a computer system that uses microcode subroutines to execute complex macroinstructions. Each macroinstruction is used to index a table. Simple macro instructions have a single microinstruction counterpart in the table, and such microinstruction is performed directly to execute that macroinstruction. The table entry

corresponding to more complex macroinstructions is a jump microinstruction, with the target of the microcode jump being an appropriate subroutine in a microcode memory. (Abstract) Bosshart notes that the microcode of a computer system is generally not available directly to the programmer and in some systems, such as the Explorer symbolic computer, the microcode memory is actually a RAM which is loaded with the actual microinstructions at system boot-up time. This allows the processing system to be configured in different ways, and optimized for the particular task at hand. However, the microcode is still not directly accessible to application programs running on the system. (col. 2, line 64 - col. 5, line 5) Bosshart's microcode memory (16) is coupled to the central processor (10) and can be an independent RAM which is loaded with microcode at system boot-up time, or it can be ROM or EPROM which is built into the central processor chip. Also coupled to the central processor (10) is a macrocode table (18), which is a memory preferably of the same type as the microcode memory (16). In fact, as described with respect to Bosshart's Figures 7 and 8, the macrocode table (18) may be nothing more than a reserved block within the microcode memory (16). (col. 3, lines 19-32) In addition, Bosshart describes the macrocode table (18) itself as a table of microinstructions, some macroinstructions therein mapping to a single microinstruction, so called "macromode instructions," and others mapping to a JUMP or DISPATCH instruction giving the address of the appropriate microcode routine as the destination of the JUMP or DISPATCH. (col. 3, line 62 - col. 4, line37)

Clearly, Bosshart teaches what Applicant has referred to as a control ROM in the instant application. Under Bosshart's teaching, microcode memory is where microcode routines are stored for the execution of complex operations that require execution of more than one microinstruction. In addition, Bosshart's microcode memory and macrocode table are loaded at system boot-up time, thus not providing the capability for direct access by application programs running on the system. In that the macrocode table contains the target address for the JUMP or DISPATCH instructions, it follows then that the target address is not directly accessible by application programs. A resource that can be directly accessed by an application program is what is commonly referred to in the art as an "architectural" resource; a resource that cannot be accessed directly by an application

program is known as a "native" resource. Accordingly, since Bosshart's macrocode table and microcode memories cannot be directly accessed by application programs, they are thus native resources, and the inability to access these resources is a limitation of the art that is addressed by the instant invention.

Applicant's invention, in contrast to Bosshart's invention, allows a microprocessor to execute application programs directly from memory, where the application programs are coded using native micro instructions. The instant application teaches a microprocessor apparatus that allows native instructions, retrieved from memory, to bypass macro instruction decoding functions, thereby allowing the explicit prescription of native resources. For example, according to Bosshart's teaching, the target address for the JUMP or DISPATCH instructions is not directly accessible by an application program. According to Applicant's invention, specification of a target address for an unconditional jump native instruction (which is produced by translation of a native branch macro instruction) is provided by loading the target address in an architectural register (executing a MOV instruction to load register EAX in an x86-compatible embodiment) prior to executing the native branch macro instruction.

Amended claim 1 is provided below for ease of reference.

1. An apparatus in a microprocessor for executing programmed native instructions that are provided directly to the microprocessor via an external instruction bus, the apparatus comprising:

instruction translation logic, configured to retrieve macro instructions provided via the external instruction bus, and configured to decode each of said macro instructions into associated native instructions for execution by the microprocessor, wherein said instruction translation logic decodes a native bypass macro instruction into an unconditional jump native instruction directing that program control be transferred to a memory address containing the programmed native instructions, and wherein said memory address is explicitly prescribed by contents of an architectural register, said contents and said architectural register being prescribed by a macro instruction; and

bypass logic, coupled to said instruction translation logic, configured to disable said instruction translation logic upon detection of said native bypass macro instruction, and configured to provide the programmed native instructions for execution by the microprocessor, thereby bypassing said instruction translation logic.

In her rejection of claim 1, the Examiner noted that Bosshart has taught an apparatus in a microprocessor for executing programmed native instructions that are provided directly to the microprocessor via an external instruction bus, the apparatus comprising:

a. instruction translation logic, configured to retrieve macro instructions provided via the external instruction bus, and configured to decode each of said macro instructions into associated native instructions for execution by the microprocessor, where said instruction translation logic decodes a native bypass macro instruction into an unconditional jump native instruction directing that program control be transferred to a memory address containing the programmed native instructions; and

b. bypass logic, coupled to said instruction translation logic, configured to disable said instruction translation logic upon detection of said native bypass macro instruction, and configured to provide the programmed native instructions for execution by the microprocessor, thereby bypassing said instruction translation logic.

Applicant respectfully disagrees with the Examiner's rejection of claim 1 and notes that claim 1 recites, in combination with other elements and limitations, "wherein said memory address is explicitly prescribed by contents of an architectural register, said contents and said architectural register being prescribed by a macro instruction." The ability to prescribe, by a macro instruction, the memory address to which program control is transferred for execution of the programmed native instructions is a feature that distinguishes Applicant's invention over the prior art. Applicant has searched the teachings of Bosshart and cannot find any reference to the ability to prescribe or otherwise specify the target address for a native instruction sequence loaded in memory by direct access by an application program. What Applicant finds is that Bosshart's invention is directly towards providing the ability for more complex operations prescribed by macroinstructions to be performed by directing control to a microcode sequence that has be loaded into a control ROM (or RAM). Thus, Bosshart addresses a different problem than Applicant. Applicant's invention addresses, among other things, the inability to directly specify native resources within an application program.

For these reasons, Applicant respectfully requests that the rejection of claim 1 be withdrawn.

With respect to claims 2-10, these claims depend from claim 1 and add further limitations that are neither anticipated nor made obvious by Bosshart. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 2-10.

Claim 11 is recited below for ease of reference.

11. An apparatus, for allowing a micro instruction to be directly provided from an external instruction bus to execution logic within a pipeline microprocessor, the apparatus comprising:

a translator, for receiving macro instructions from a macro instruction bus, and for translating each of said macro instructions into associated micro instructions, said associated micro instructions being provided to the execution logic via a micro instruction bus, wherein said translator translates a native bypass macro instruction into an unconditional jump native instruction directing that program control be transferred to a memory address containing the micro instruction, and wherein said memory address is explicitly prescribed by contents of an architectural register, said contents and said architectural register being prescribed by a macro instruction; and

bypass logic, coupled to said translator, for routing the micro instruction to the execution logic, said bypass logic comprising:

a mode detector, for detecting said native bypass macro instruction, and for directing that said translator cease instruction translation; and

native instruction routing logic, coupled to said mode detector, for receiving said micro instruction from said macro instruction bus, and for providing said micro instruction to said micro instruction bus, thereby circumventing said translator.

In rejection of claim 11, the Examiner stated that Bosshart has taught an apparatus, for allowing a micro instruction to be directly provided from an external instruction bus to execution logic within a pipeline microprocessor, the apparatus comprising:

a translator, for receiving macro instructions from a macro instruction bus, and for translating each of said macro instructions into associated micro instructions, said associated micro instructions being provided to the execution logic via a micro instruction bus, wherein said translator translates a native bypass macro instruction into an unconditional jump native instruction directing that program control be transferred to a memory address containing the micro instruction; and

bypass logic, coupled to said translator, for routing the micro instruction to the execution logic, said bypass logic comprising:

a mode detector, for detecting said native bypass macro instruction, and for directing that said translator cease instruction translation; and

native instruction routing logic, coupled to said mode detector, for receiving said micro instruction from said macro instruction bus, and for providing said micro instruction to said micro instruction bus, thereby circumventing said translator.

Applicant respectfully disagrees with the Examiner's rejection of claim 11 and notes that claim 11, in similar fashion to claim 1, recites, "wherein said memory address is explicitly prescribed by contents of an architectural register, said contents and said architectural register being prescribed by a macro instruction." And as noted above with reference to the rejection of claim 1, providing for prescription of the memory address by a macro instruction patentably distinguishes Applicant's invention over the prior art. Bosshart fails to teach prescription of a target address for a native instruction sequence loaded in memory by direct access by an application program. Furthermore, Bosshart fails to provide any suggestion or other hint that would motivate one skilled in the art to provide for prescription of the target address. Bosshart's target addresses are fixed at system boot-up time and cannot be accessed by an application program.

For these reasons, Applicant respectfully requests that the rejection of claim 11 be withdrawn.

With respect to claims 12-17, these claims depend from claim 11 and add further limitations that are neither anticipated nor made obvious by Bosshart. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 12-17.

Claim 18 is repeated below for ease of reference.

18. A microprocessor for executing micro instructions directly from memory, the microprocessor comprising:

translation logic, for receiving macro instructions from the memory, and for decoding said macro instructions into corresponding micro instructions for execution by the microprocessor;

mode detection logic, coupled to said translation logic, for detecting bypass macro instructions, and for directing the microprocessor to execute the micro instructions directly from the memory rather than via said translation logic, said bypass macro instructions comprising:

a native branch macro instruction, directing that program control be transferred to a target address, wherein said translation logic decodes said native branch macro instruction into an unconditional jump native instruction directing that program control be transferred to said target address, and wherein said target address contains the micro instructions, and wherein said target address is explicitly prescribed by contents of an architectural register, said contents and said architectural register being prescribed by a macro instruction; and

a native branch return macro instruction, directing that program control be transferred to a return address; and

an instruction router, coupled to said mode detection logic, for receiving the micro instructions, and for routing the micro instructions to execution logic, thereby bypassing said translation logic.

With reference to the rejection of claim 18, the Examiner noted that Bosshart has taught a microprocessor for executing micro instructions directly from memory, the microprocessor comprising:

translation logic, for receiving macro instructions from the memory, and for decoding said macro instructions into corresponding micro instructions for execution by the microprocessor;

mode detection logic, coupled to said translation logic, for detecting bypass macro instructions, and for directing the microprocessor to execute the micro instructions directly from the memory rather than via said translation logic, said bypass macro instructions comprising:

a native branch macro instruction, directing that program control be transferred to a target address, wherein said translation logic decodes said native branch macro instruction into an unconditional jump native instruction directing that program control be transferred to said target address, and

a native branch return macro instruction, directing that program control be transferred to a return address; and

an instruction router, coupled to said mode detection logic, for receiving the micro instructions, and for routing the micro instructions to execution logic, thereby bypassing said translation logic.

Applicant respectfully disagrees with the rejection of claim 18 and notes that, like claims 1 and 11, claim 18 recites "wherein said target address is explicitly prescribed by contents of an architectural register, said contents and said architectural register being prescribed by a macro instruction." And as argued above in disputation of the rejections of claims 1 and 11, Applicant points out that Bosshart's teaching is not directed towards providing for explicit prescription of a target address by contents of an architectural register. Bosshart's target address cannot be accessed by an application program.

Accordingly, Applicant respectfully requests that the Examiner withdraw his rejection of claim 18.

With respect to claims 19 and 21-24, these claims depend from claim 18 and add further limitations that are neither anticipated nor made obvious by Bosshart. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 19 and 21-24.

CONCLUSIONS

In view of the arguments advanced above, Applicant respectfully submits that claims 1-19 and 21-24 are in condition for allowance. Reconsideration of the rejections is requested, and allowance of the claims is solicited.

Applicant earnestly requests that the Examiner contact the undersigned practitioner by telephone if the Examiner has any questions or suggestions concerning this amendment, the application, or allowance of any claims thereof.

EXPRESS MAIL LABEL NUMBER: EO 002 118 563 US
DATE OF DEPOSIT: 6/23/2004
I hereby certify that this paper is being deposited with the U.S. Postal Service Express Mail Post Office to Addressee Service under 37 C.F.R. §1.10 on the date shown above and is addressed to Mail Stop RCE, Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450.

Respectfully submitted,

HUFFMAN PATENT GROUP, LLC

By _____
RICHARD K. HUFFMAN
Reg. No. 41,082
Tel.: (719) 575-9998

Date: 6/23/04